

Amendments to the Claims

1 Claim 16 (currently amended): A computer implemented method for programmatically creating a
2 distributed object program in which at least one complex object is passed as a parameter, wherein
3 the programmatically-created program is programmatically generated from a programmer-written
4 program which is not specially adapted for distributed execution, the distributing one or more
5 objects of a program across more than one physical device, each object containing one or more
6 programmed member functions, said member functions having complex objects, said complex
7 objects including one or more programmed member functions, as parameters, said method
8 comprising the computer executable steps of:

9 identifying all of the one or more objects in the programmer-written program, wherein
10 each of the objects contains one or more programmed member functions and wherein at least one
11 of the programmed member functions is written to pass one of the objects as a parameter;

12 determining a first set which of the identified objects which are to reside on a first
13 computer and a second set which of the identified objects which are to reside on a second
14 computer, wherein the first set and the second set together comprise the identified objects of the
15 programmer-written program and the first set and the second set each include at least one of the
16 identified objects; such that the distributed system will consist of at least a first object on a first
17 computer and a second object on a second computer;

18 — identifying all programmed methods contained in each object that may be accessed from a
19 remote computer;

20 programmatically generating, upon detecting that a first object in the first set contains
21 logic to call one of the programmed member functions of a second object in the second set, a first

22 proxy and a second proxy for each the second object, wherein the first proxy is generated to be
23 installed on the first computer and the second proxy is generated to be installed on the second
24 computer;

25 programmatically generating logic in the first proxy that will programmatically generate a
26 third proxy, responsive to detecting that the call to the programmed member function of the
27 second object will pass, as a parameter, a third object that is a complex object and that is one of
28 the objects in the first set, wherein the third proxy is generated to be installed on the first
29 computer; and

30 programmatically generating logic in the second proxy that will programmatically generate
31 a fourth proxy, responsive to a call from the first proxy that includes a reference to the third
32 proxy, wherein the fourth proxy is generated to be installed on the second computer,

33 such that, at run time, the first object can transparently access the programmed member
34 function of the second object and the programmed member function of the second object can
35 transparently access a programmed member function of the third object, that may be accessed
36 from a remote computer, said first proxy residing on said first computer and said second proxy
37 residing on said second computer, said first proxy containing network linkage and indication to
38 access programmed member functions on said second proxy on said second computer including
39 logic to transfer and translate complex objects which reside on said first computer used as
40 member function parameters and said second proxy containing linkage and indication to access
41 said programmed member functions on said second object including logic to transfer and translate
42 complex objects, said complex objects containing one or more programmed member functions
43 and reside on said first computer, used as member function parameters; and,

44 —— accessing said remote programmed methods through said proxies.

1 Claim 17 (currently amended): A method The method as claimed in Claim 16, wherein:
2 said the logic in said first proxy further comprises programmatically-generated logic to on
3 said first computer to transfer and translate complex data objects comprising the steps of:
4 —— creating a third proxy, for said complex object, which is to reside on said first computer
5 with said complex object, said third proxy containing linkage and indication to access
6 programmed member functions on said complex object;
7 —— creating (1) create a reference table entry which correlates said the third proxy
8 object to said complex the third object, which may be accessed by said the third proxy object to
9 access said complex when invoking programmed member functions of the third object; (2)
10 translate calls for the programmed member function of the second object that are received from
11 the first object and that pass the third object as a parameter, whereby a reference to the third
12 proxy replaces the third object on the received calls, and forward the translated calls to the second
13 proxy; and (3) upon receiving, from the second proxy, responses to the translated calls, return the
14 responses to the first object; ;and,
15 —— passing as a member function parameter to said second proxy on said second
16 machine a reference to said third proxy, in place of said complex object when said complex
17 object is to be a parameter in a member function call to said second object on said second
18 machine.
19 said logic in said the second proxy further comprises programmatically-generated logic to
20 on said second computer to transfer and translate complex data objects comprising the steps of:

21 _____ creating a fourth proxy for said complex object on said first computer which is to
22 reside on said second computer, said fourth proxy containing network linkage and indication
23 necessary to access programmed member functions on said third proxy on said first machine;
24 _____ creating a (1) create a second reference table entry which correlates said the fourth
25 proxy to a to the reference to said the third proxy on said third computer, which may be accessed
26 by said the fourth proxy to access said when forwarding calls to the third proxy; (2) invoke the
27 programmed member function of the second object, responsive to receiving one of the translated
28 calls that is forwarded from the first proxy, wherein the second reference table entry is consulted
29 so that an indication of the fourth proxy is substituted, on the invocation, for the reference to the
30 third proxy; and (3) upon receiving, from the programmed member function of the second object,
31 a response to the forwarded call, return the response to the first proxy;

32 the fourth proxy further comprises programmatically-generated logic to (1) consult the
33 second reference table entry, responsive to receiving a call from the programmed member
34 function of the second object, thereby determining that the received call corresponds to the third
35 proxy; (2) translate the call received from the programmed member function of the second object
36 such that the translated call refers to the third proxy and forward the translated call to the third
37 proxy; and (3) upon receiving, from the third proxy, a response to the translated call, return the
38 response to the programmed member function of the second object; and

39 the third proxy further comprises programmatically-generated logic to (1) consult the
40 reference table entry, responsive to receiving the translated call from the fourth proxy, thereby
41 determining that the received translated call corresponds to the third object; (2) translate the call
42 received from the fourth proxy to invoke the programmed member function of the third object

43 and forward the translated call to the third object, where the programmed member function of the
44 third object will then be executed; and (3) upon receiving, from the programmed member function
45 of the third object, a result of the execution, return the result to the fourth proxy. ;

46 ————— passing as a member function parameter to said second object from said second
47 proxy on said second computer an indication of said fourth proxy, in place of said reference to
48 said third proxy on said first computer, which represents said complex object on said first
49 computer.

50 ————— said network linkage and indication in said fourth proxy necessary to access programmed
51 member functions on said third proxy on said first computer comprising the steps of:

52 ————— looking up said fourth proxy in said reference table on said second computer to
53 determine which object on said first machine said fourth object is a proxy for, said lookup
54 returning a reference to said third proxy on said first computer;

55 ————— calling the appropriate programmed member functions in said third proxy on said
56 first computer.

57 ————— said linkage and indication in said third proxy necessary to access programmed methods
58 on said complex object comprising the steps of:

59 ————— looking up said third proxy in said reference table on said first computer to
60 determine which object on said first machine said third object is a proxy for, said lookup returning
61 a reference to said complex object on said first computer;

62 ————— calling the appropriate programmed member functions in said complex object.

1 Claim 18 (currently amended): A method The method as claimed in Claim 17 wherein 16.

2 wherein one of said complex objects the third object is said identical to the first object, such that
3 the call to the programmed member function of the second object executes as a callback on the
4 first object, on said first computer.

1 Claim 19 (currently amended): A method The method as claimed in Claim 16, wherein 17
2 wherein said the reference table entry and the second reference table entry are created as is a
3 database entries.

1 Claim 20 (currently amended): A computer program product for programmatically creating a
2 distributed object program in which at least one complex object is passed as a parameter, wherein
3 the programmatically-created program is programmatically generated from a programmer-written
4 program which is not specially adapted for distributed execution, the distributing one or more
5 objects of a program across more than one physical device, each object containing one or more
6 programmed member functions, said member functions having complex objects, said complex
7 objects including one or more programmed member functions, as parameters, said computer
8 program product comprising:

9 a computer-readable storage medium have computer-readable program code means
10 embodied in said medium, said computer-readable program code means comprising:

11 computer-readable program code means for identifying all of the one or more
12 objects in the programmer-written program, wherein each of the objects contains one or more
13 programmed member functions and wherein at least one of the programmed member functions is
14 written to pass one of the objects as a parameter;

15 computer-readable program code means for determining a first set which of the
16 identified objects which are to reside on a first computer and which a second set of the identified
17 objects which are to reside on a second computer, wherein the first set and the second set
18 together comprise the identified objects of the programmer-written program and the first set and
19 the second set each include at least one of the identified objects; such that the distributed system
20 will consist of at least a first object on a first computer and a second object on a second computer;
21 _____ computer-readable program code means for identifying all programmed methods
22 contained in each object that may be accessed from a remote computer;

23 computer-readable program code means for programmatically generating, upon
24 detecting that a first object in the first set contains logic to call one of the programmed member
25 functions of a second object in the second set, a first proxy and a second proxy for each the
26 second object, wherein the first proxy is generated to be installed on the first computer and the
27 second proxy is generated to be installed on the second computer;

28 computer-readable program code means for programmatically generating logic in
29 the first proxy that will programmatically generate a third proxy, responsive to detecting that the
30 call to the programmed member function of the second object will pass, as a parameter, a third
31 object that is a complex object and that is one of the objects in the first set, wherein the third
32 proxy is generated to be installed on the first computer; and

33 computer-readable program code means for programmatically generating logic in
34 the second proxy that will programmatically generate a fourth proxy, responsive to a call from the
35 first proxy that includes a reference to the third proxy, wherein the fourth proxy is generated to be
36 installed on the second computer,

37 such that, at run time, the first object can transparently access the programmed member
38 function of the second object and the programmed member function of the second object can
39 transparently access a programmed member function of the third object, that may be accessed
40 from a remote computer, said first proxy residing on said first computer and said second proxy
41 residing on said second computer, said first proxy containing network linkage and indication to
42 access programmed member functions on said second proxy on said second computer including
43 logic to transfer and translate complex objects which reside on said first computer used as
44 member function parameters and said second proxy containing linkage and indication to access
45 said programmed member functions on said second object including logic to transfer and translate
46 complex objects, said complex objects containing one or more programmed member functions
47 and reside on said first computer, used as member function parameters; and,
48 computer-readable program code means for accessing said remote programmed
49 methods through said proxies.

1 Claim 21 (currently amended): ~~A computer~~ The computer program product as claimed in Claim
2 20, wherein:

3 said logic in said the first proxy on said first computer to transfer and translate complex
4 data objects comprising the steps of:
5 creating a third proxy, for said complex object, which is to reside on said first
6 computer with said complex object, said third proxy containing linkage and indication to access
7 programmed member functions on said complex object;
8 creating further comprises programmatically-generated logic to (1) create a

9 reference table entry which correlates said the third proxy object to said complex the third object,
10 which may be accessed by said the third proxy object to access said complex when invoking
programmed member functions of the third object; (2) translate calls for the programmed member
11 function of the second object that are received from the first object and that pass the third object
12 as a parameter, whereby a reference to the third proxy replaces the third object on the received
13 calls, and forward the translated calls to the second proxy; and (3) upon receiving, from the
14 second proxy, responses to the translated calls, return the responses to the first object; ; and;
15 passing as a member function parameter to said second proxy on said second
16 machine a reference to said third proxy, in place of said complex object when said complex
17 object is to be a parameter in a member function call to said second object on said second
18 machine.

19 — said logic in said the second proxy on said second computer to transfer and translate
20 complex data objects comprising the steps of:

21 — creating a fourth proxy for said complex object on said first computer which is to
22 reside on said second computer, said fourth proxy containing network linkage and indication
23 necessary to access programmed member functions on said third proxy on said first machine;

24 — creating a further comprises programmatically-generated logic to (1) create a
25 second reference table entry which correlates said the fourth proxy to a to the reference to said
26 the third proxy on said third computer, which may be accessed by said the fourth proxy to access
27 said when forwarding calls to the third proxy; (2) invoke the programmed member function of the
28 second object, responsive to receiving one of the translated calls that is forwarded from the first
29 proxy, wherein the second reference table entry is consulted so that an indication of the fourth

30 proxy, wherein the second reference table entry is consulted so that an indication of the fourth

31 proxy is substituted, on the invocation, for the reference to the third proxy; and (3) upon
32 receiving, from the programmed member function of the second object, a response to the
33 forwarded call, return the response to the first proxy;

34 the fourth proxy further comprises programmatically-generated logic to (1) consult the
35 second reference table entry, responsive to receiving a call from the programmed member
36 function of the second object, thereby determining that the received call corresponds to the third
37 proxy; (2) translate the call received from the programmed member function of the second object
38 such that the translated call refers to the third proxy and forward the translated call to the third
39 proxy; and (3) upon receiving, from the third proxy, a response to the translated call, return the
40 response to the programmed member function of the second object; and

41 the third proxy further comprises programmatically-generated logic to (1) consult the
42 reference table entry, responsive to receiving the translated call from the fourth proxy, thereby
43 determining that the received translated call corresponds to the third object; (2) translate the call
44 received from the fourth proxy to invoke the programmed member function of the third object
45 and forward the translated call to the third object, where the programmed member function of the
46 third object will then be executed; and (3) upon receiving, from the programmed member function
47 of the third object, a result of the execution, return the result to the fourth proxy. ~

48 passing as a member function parameter to said second object from said second
49 proxy on said second computer an indication of said fourth proxy, in place of said reference to
50 said third proxy on said first computer, which represents said complex object on said first
51 computer.

52 said network linkage and indication in said fourth proxy necessary to access programmed

53 member functions on said third proxy on said first computer comprising the steps of:
54 ————— looking up said fourth proxy in said reference table on said second computer to
55 determine which object on said first machine said fourth object is a proxy for, said lookup
56 returning a reference to said third proxy on said first computer;
57 ————— calling the appropriate programmed member functions in said third proxy on said
58 first computer;
59 ————— said linkage and indication in said third proxy necessary to access programmed methods
60 on said complex object comprising the steps of:
61 ————— looking up said third proxy in said reference table on said first computer to
62 determine which object on said first machine said third object is a proxy for, said lookup returning
63 a reference to said complex object on said first computer;
64 ————— calling the appropriate programmed member functions in said complex object.

1 Claim 22 (currently amended): A computer The computer program product as claimed in Claim
2 21 wherein one of said complex objects 20, wherein the third object is said identical to the first
3 object, such that the call to the programmed member function of the second object executes as a
4 callback on the first object, on said first computer.

1 Claim 23 (currently amended): A computer The computer program product as claimed in Claim
2 21 wherein said 20, wherein the reference table is a entry and the second reference table entry are
3 created as database entries.

1 Claim 24 (currently amended): A computer system for programmatically creating a distributed
2 object program in which at least one complex object is passed as a parameter, wherein the
3 programmatically-created program is programmatically generated from a programmer-written
4 program which is not specially adapted for distributed execution, the distributing one or more
5 objects of a program across more than one physical device, each object containing one or more
6 programmed member functions, said member functions having complex objects, said complex
7 objects including one or more programmed member functions, as parameters, said system
8 comprising:

9 means for identifying all of the one or more objects in the programmer-written program,
10 wherein each of the objects contains one or more programmed member functions and wherein at
11 least one of the programmed member functions is written to pass one of the objects as a
12 parameter;

13 means for determining which a first set of the identified objects which are to reside on a
14 first computer and a second set which of the identified objects which are to reside on a second
15 computer, wherein the first set and the second set together comprise the identified objects of the
16 programmer-written program and the first set and the second set each include at least one of the
17 identified objects; such that the distributed system will consist of at least a first object on a first
18 computer and a second object on a second computer;

19 means for identifying all programmed methods contained in each object that may be
20 accessed from a remote computer;

21 means for programmatically generating, upon detecting that a first object in the first set
22 contains logic to call one of the programmed member functions of a second object in the second

23 set, a first proxy and a second proxy for each the second object, wherein the first proxy is
24 generated to be installed on the first computer and the second proxy is generated to be installed
25 on the second computer;

26 means for programmatically generating logic in the first proxy that will programmatically
27 generate a third proxy, responsive to detecting that the call to the programmed member function
28 of the second object will pass, as a parameter, a third object that is a complex object and that is
29 one of the objects in the first set, wherein the third proxy is generated to be installed on the first
30 computer; and

31 means for programmatically generating logic in the second proxy that will
32 programmatically generate a fourth proxy, responsive to a call from the first proxy that includes a
33 reference to the third proxy, wherein the fourth proxy is generated to be installed on the second
34 computer,

35 such that, at run time, the first object can transparently access the programmed member
36 function of the second object and the programmed member function of the second object can
37 transparently access a programmed member function of the third object, that may be accessed
38 from a remote computer, said first proxy residing on said first computer and said second proxy
39 residing on said second computer, said first proxy containing network linkage and indication to
40 access programmed member functions on said second proxy on said second computer including
41 logic to transfer and translate complex objects which reside on said first computer used as
42 member function parameters and said second proxy containing linkage and indication to access
43 said programmed member functions on said second object including logic to transfer and translate
44 complex objects, said complex objects containing one or more programmed member functions

45 and reside on said first computer, used as member function parameters; and,
46 ——— means for accessing said remote programmed methods through said proxies.

1 Claim 25 (currently amended): A system The system as claimed in Claim 24, wherein:
2 said logic in said the first proxy on said first computer to transfer and translate complex
3 data objects comprising the steps of:
4 ——— creating a third proxy, for said complex object, which is to reside on said first
5 computer with said complex object, said third proxy containing linkage and indication to access
6 programmed member functions on said complex object;
7 ——— creating further comprises programmatically-generated logic to (1) create a
8 reference table entry which correlates said the third proxy object to said complex the third object,
9 which may be accessed by said the third proxy object to access said complex when invoking
10 programmed member functions of the third object; (2) translate calls for the programmed member
11 function of the second object that are received from the first object and that pass the third object
12 as a parameter, whereby a reference to the third proxy replaces the third object on the received
13 calls, and forward the translated calls to the second proxy; and (3) upon receiving, from the
14 second proxy, responses to the translated calls, return the responses to the first object; and,
15 ——— passing as a member function parameter to said second proxy on said second
16 machine a reference to said third proxy, in place of said complex object when said complex
17 object is to be a parameter in a member function call to said second object on said second
18 machine;
19 said logic in said the second proxy on said second computer to transfer and translate

20 complex data objects comprising the steps of:

21 ————— creating a fourth proxy for said complex object on said first computer which is to
22 reside on said second computer, said fourth proxy containing network linkage and indication
23 necessary to access programmed member functions on said third proxy on said first machine;
24 ————— creating a further comprises programmatically-generated logic to (1) create a
25 second reference table entry which correlates said the fourth proxy to-a to the reference to said
26 the third proxy on said third computer, which may be accessed by said the fourth proxy to-access
27 said when forwarding calls to the third proxy; (2) invoke the programmed member function of the
28 second object, responsive to receiving one of the translated calls that is forwarded from the first
29 proxy, wherein the second reference table entry is consulted so that an indication of the fourth
30 proxy is substituted, on the invocation, for the reference to the third proxy; and (3) upon
31 receiving, from the programmed member function of the second object, a response to the
32 forwarded call, return the response to the first proxy.

33 the fourth proxy further comprises programmatically-generated logic to (1) consult the
34 second reference table entry, responsive to receiving a call from the programmed member
35 function of the second object, thereby determining that the received call corresponds to the third
36 proxy; (2) translate the call received from the programmed member function of the second object
37 such that the translated call refers to the third proxy and forward the translated call to the third
38 proxy; and (3) upon receiving, from the third proxy, a response to the translated call, return the
39 response to the programmed member function of the second object; and

40 the third proxy further comprises programmatically-generated logic to (1) consult the
41 reference table entry, responsive to receiving the translated call from the fourth proxy, thereby

42 determining that the received translated call corresponds to the third object; (2) translate the call
43 received from the fourth proxy to invoke the programmed member function of the third object
44 and forward the translated call to the third object, where the programmed member function of the
45 third object will then be executed; and (3) upon receiving, from the programmed member function
46 of the third object, a result of the execution, return the result to the fourth proxy.

47 ————— passing as a member function parameter to said second object from said second
48 proxy on said second computer an indication of said fourth proxy, in place of said reference to
49 said third proxy on said first computer, which represents said complex object on said first
50 computer.

51 ————— said network linkage and indication in said fourth proxy necessary to access programmed
52 member functions on said third proxy on said first computer comprising the steps of:

53 ————— looking up said fourth proxy in said reference table on said second computer to
54 determine which object on said first machine said fourth object is a proxy for, said lookup
55 returning a reference to said third proxy on said first computer;

56 ————— calling the appropriate programmed member functions in said third proxy on said
57 first computer;

58 ————— said linkage and indication in said third proxy necessary to access programmed methods
59 on said complex object comprising the steps of:

60 ————— looking up said third proxy in said reference table on said first computer to
61 determine which object on said first machine said third object is a proxy for, said lookup returning
62 a reference to said complex object on said first computer;

63 ————— calling the appropriate programmed member functions in said complex object.

1 Claim 26 (currently amended): A system The system as claimed in Claim 25 wherein one of said
2 complex objects 24, wherein the third object is identical to the said first object, such that the call
3 to the programmed member function of the second object executes as a callback on the first
4 object, on said first computer.

1 Claim 27 (currently amended): A system The system as claimed in Claim 25 wherein said 24,
2 wherein the reference table entry and the second reference table entry are created as is a database
3 entries.